SMART WASTE MANAGEMENT SYSTEM

Group 09: Nicci Sequeira, Priyadarshini Krishna Shobha, and Runeet Palekar

Service Computing Department, IAAS, University of Stuttgart
st169712@stud.uni-stuttgart.de
st169614@stud.uni-stuttgart.de
st169659@stud.uni-stuttgart.de

Abstract. The 21st century has witnessed an increase in the generation of waste across cities because of humans flooding into big cities for opportunities followed by indulgence into urban lifestyle and recreational activities on a regular basis. The unmanageable wastes, if not disposed correctly can pose serious threat to health and environment. It is therefore crucial to separate waste forms at the source for recycling purposes since mixing dry and wet waste, breaks down in landfills to create nasty greenhouse gases and this is considered to be valueless. Additionally, many in the society lack awareness regarding the nature of waste and its disposal measures thus making waste management a mammoth task to be achieved. Small cities are equally affected due to financial constraints and absence of areas for landfills and incineration. We furthermore need to focus towards making our waste practices more sustainable with the involvement of citizens, public and private sectors. Recently, outdated methods of waste collection are progressively being replaced with sensor-enabled bins thus showing the impact of IoT and a promising future. Such smart waste management solutions help create higher efficiency in terms of resources and costs associated with keeping cities clean.

Keywords: Waste management · dustbins · sensors · IoT · environment · timely disposal · MQTT

1 System Introduction

1.1 Project Scope:

Waste management is one of the priority issues of the century and so opting for a smart solution will be significant for the growth of the human civilization and will also help in building smart cities aiming to improve the quality of life. The project will not only help us with timely disposal of waste to but also help create a level of understanding among the waste management authorities who will be involved in collecting waste efficiently from a given area. This results in multiple, unwanted trips being avoided and diversion of attention to other areas where waste collection is needed thus optimizing garbage collection trips. Inshort, collection and analysis of area specific data within or across buildings on waste volumes will help in better planning. Additionally, by observing how often

the system is used, we could add-on to the existing setup. From a citizen's point of view, these smart dustbins will help raise awareness regarding sanitation, type of waste and its disposal measures. The waste segregation that we will focus on is dry and wet waste. Examples for each of the waste types are : Dry waste - wooden sticks, metal cans, glass bottles, paper, cardboard, plastic cutlery which can be reused/recycled and Wet waste - vegetable and fruit peels and pieces, tea bags, eggshells, bones and entrails, cooked food remains.

1.2 Project Goal:

The objective of the project is to create an automated waste management system that involves isolation of wastes into respective dustbins based on its nature, monitoring the level or quantity of garbage within each of the dustbins followed by collection and disposal of waste by concerned authorities through a notification system at the commercial level [13]. It serves the purpose of optimizing the functionality of a typical waste disposal unit by reducing human involvement and smartly managing the operations. Such systems can be implemented in eateries within offices wherein the remains can be dropped into a bin. Certain thresholds are set for each dustbins in order to check the level of garbage to avoid overflow and release of unpleasant odour. Accordingly, it will notify the respective authorities through a messaging application for collection and timely disposal of the waste to maintain a clean and healthy environment. Pick-up schedules can therefore be organized only when it is required rather than collecting waste daily at some predefined time-slot. A user interface web application will help the authorities with real time monitoring of garbage level in each dustbins along with a timely trend to understand the waste generation demographics and system usage by users that assist them to take appropriate steps to enhance the existing system.

2 System Analysis

Smart waste management plays a major role in building a smart city. The segregation of waste into dry and wet waste plays a vital role in waste management. System analysis serves the purpose of examining a system or its parts in order to identify its objectives and ensures that all the components of the system work efficiently to accomplish its motives.

2.1 Requirements:

Following are the requirements for a smart waste management project:

1. Fetching sensor details connected to arduino :
   - PIR motion sensor : It will be the starting point of the system since the rest of the operations are followed sequentially.
   - Ultrasonic sensor : It is used to compute the total length of the dustbin with and without trash to determine the threshold needed for event trigger.
   - Load cell : The sensor is associated with wet waste since organic waste usually are heavier in weight due to dampness when compared to dry waste.
2. Actuation performed by the system on the basis of AI planning :
   - Servo Motor : Servo motor is actuated accordingly to reach a defined goal state.
   - LED : It acts as an indication component.
3. Publishing of sensor data and retrieving relevant information from JSON file.
4. Information and signal exchange (publish-subscribe) via MQTT broker.
5. Sending a notification to the supervisor about the dustbins being full for timely collection.
6. Graphical User Interface (GUI) for monitoring and manual intervention when required.

The above-mentioned requirements can be classified into functional and non-functional requirements along with its priorities through a table representation below:

Table 1. Requirements for Smart Waste Management system.

ID	Description	Type	Priority
R01	– PIR motion sensors are used to detect human presence and open the dustbin lids in order to dispose off waste. – Ultrasonic sensors detects the level of garbage within the dustbins. – Load cell estimates the mass of the garbage for convenient fetching of waste by officials without ripping the plastic.	Functional	Mandatory
R02	– Servo motor is responsible for the opening and closing of dustbins. – LED remains OFF initially and when the threshold regarding waste level within dustbins are reached, LED starts to blink as indicated by AI planner.	Functional	Mandatory
R03	JSON file at the server end keeps track of all the data received from the sensors and based on certain conditions, it sends signals to the actuators to perform tasks.	Functional	Mandatory
R04	MQTT broker acts as an indirect communication medium through which the data moves from the arduino to the server and vice versa.	Functional	Mandatory
R05	Notifications about the dustbins being full and emptied are sent to the Supervisor on Telegram.	Functional	Desirable
R06	GUI displays the waste levels for each dustbins, weight of wet dustbin and daily trends. It also performs actuation to open full dustbins since servo motor shuts the dustbins once full which can be opened by the Supervisor through GUI.	Non-Functional	Desirable

2.2 Use Case:

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). The use case diagram of this project is as shown in the Fig.1[11]. The PIR fitted dustbin's lid opens with the help of Servo motor when a user approaches it to discard the waste. The waste status is monitored and when the threshold is reached, the garbage intake to the system is stopped by shutting the lid and indicating the dustbins being full through LED blinking. Furthermore, a notification is sent to the supervisor on Telegram to keep him informed about the dustbins being full so that the disposal of waste is arranged by the concerned officials. The GUI is visible to the Supervisor only once he/she enters the correct credentials else access won't be provided thus taking care of the security and authorization aspect of the system. On logging into the GUI, it displays status of the waste in the dustbins and other choices that needs to be handled.

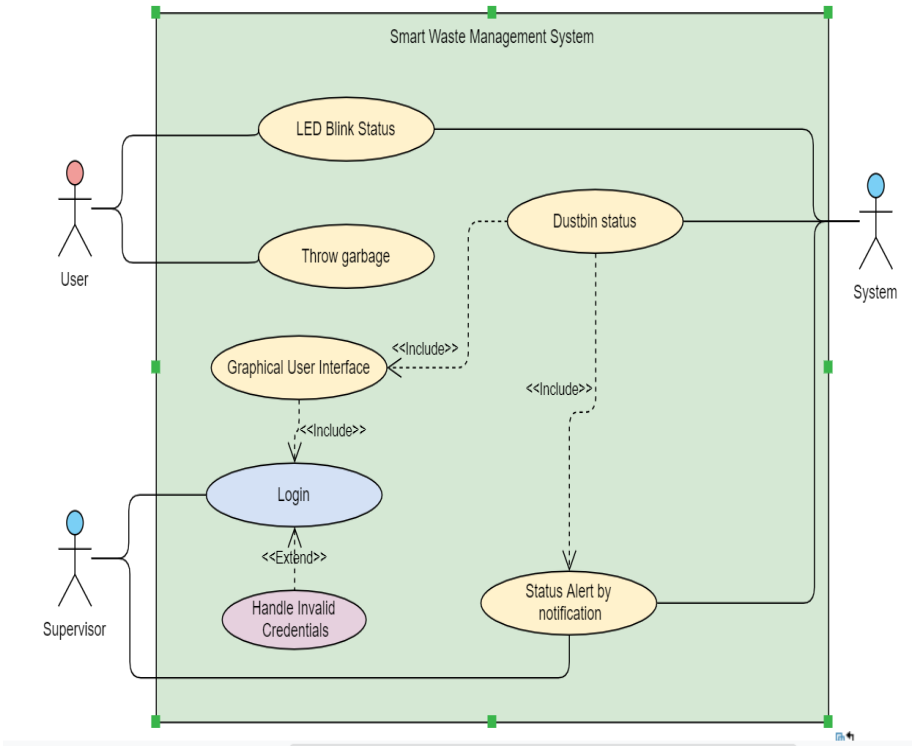

Fig. 1. Use Case diagram for Smart Waste Management System.

2.3 Data Flow diagram:

Data Flow Diagram (DFD) is a visual/graphical representation used to review the flow of data within a system and acts as an initial step in the creation of the system. Data Flow Diagram helps in better understanding of the system, identify the issues with data flow between any stages and rectify them. The data flow of our project is shown in Fig. 2. After the user disposes garbage in the system, the level of the garbage is sent to the controller unit (Server and AI planner). The controller compares the the data with threshold and performs actuation like turning ON the LED and shutting down the servo motor of the corresponding dustbin and notification will be sent to Supervisor for garbage collection. The real time data is visible to Supervisor via GUI. When the Supervisor clicks a button on GUI for garbage collection, the signal is sent to open the servo motors and the system cycle repeats itself.

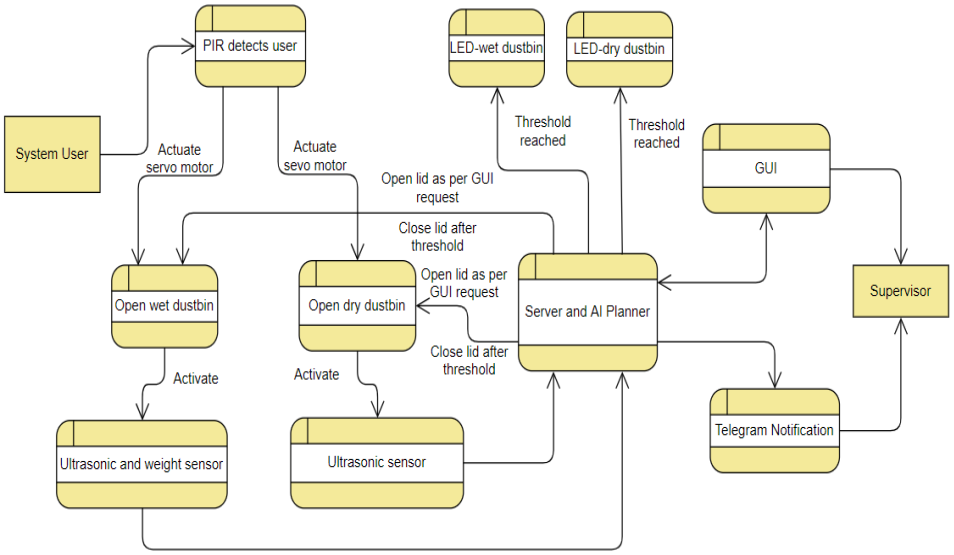

Fig. 2. Data Flow diagram for Smart Waste Management System.

3 System Architecture Design

The Internet of Things is a new kind of world in which devices equipped with embedded sensors, actuators and processors are used collaboratively linked to a network; to achieve complex tasks that require a high degree of intelligence. An architecture description is therefore a conceptual model that represents the arrangement, organized in a way that supports reasoning about the structures and behavior of the system. [9].

3.1 Architecture:

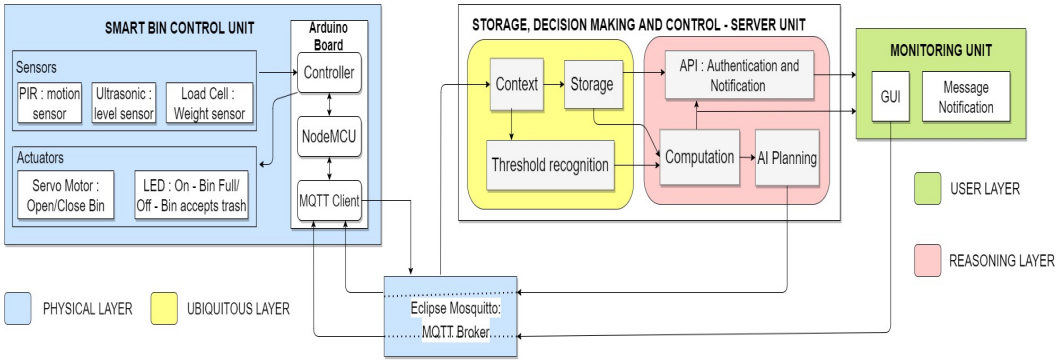

Fig. 3. Architecture of Smart Waste Management System.

The different stages in the architecture can be explained through the following layers:

– **Physical Layer:** It is responsible for sensing certain physical parameters, gathering information about the environment and provides response to the signals that drives it; all of which is connected through a network.

1. Arduino UNO - The micro-controller board acts as a medium of interaction between the digital and physical world through sensors and actuators. The sensor and actuator control algorithms are implemented using C on Arduino IDE. It can also be used for Serial communication of data with the NodeMCU.

2. PIR motion sensor (HC-SR501) - A passive infrared sensor is an electronic sensor that measures infrared (IR) light radiated/emitted/reflected from objects in its field of view. The PIR sensor has a detection angle of about 120 degrees that makes it ideal in applications where system needs to track movements across an open area.

3. Ultrasonic sensor (HC-SR04) - An ultrasonic sensor uses transducer to send(trigger signal) and receive(echo signal) ultrasonic pulses that relay back information about an object's proximity. Here, we will be detecting the pulses that are relayed back from the base of the dustbin(empty condition) and from the garbage last dumped (filling condition) to estimate the level of garbage within the trash cans.

4. Load cell (5Kg-Amplifier HX711) - It is a transducer that uses a strain gauge to measure force and output this force as an electrical signal. The load cell will be calibrated after it is attached to the wet waste bin to avoid bin weight and measure only wet trash weight collected inside the bin.

5. LED Indicator - A basic pn-junction diode which emits light when activated with a fitting voltage. It will be used to indicate the accessibility of the dustbins to end users.

6. Servo motor (9gm-SG90) - A rotary actuator that allows the control of the angular position of their motor shaft as well as the rotational speed and acceleration. It is used to control the motion of the lids of the dustbins.

7. NodeMCU (ESP8266) - A WiFi Module that is a self contained SOC with integrated TCP/IP protocol stack that provides micro-controller; access to a WiFi network. It is being used since the Arduino[8] itself does not have any wireless module in-built within it.

8. Eclipse Mosquitto - It is a MQTT (Message Queuing Telemetry Transport) broker that mediates communication among applications along with message validation and routing. It involves Paho MQTT and Pub-Sub as clients. The broker has a set of definite topics to which the system components have to either publish or subscribe for data flow through the gateway. Eg:- The AI planner and GUI subsequently publish its action plan and event to actuate the LED and servo motor respectively via the broker. It is appropriate for Internet of Things messaging (eg:-low power sensors).

– **Ubiquitous Layer:** The data collected from the sensors need to be processed in order to derive useful inferences from it. Hence in context, we represent the data in such a way that it can be communicated into the system for further reasoning (including applications). This processed sensor data is further stored in a JSON file (similar to a database) that helps to transmit data between the server and web application. The server end application produces a JSON response of the data that is received which will be used to stimulate events based on threshold conditions and data analysis. This storage unit hence contains the credentials to login to the GUI along with the

ultrasonic data for waste levels and weight information. Thresholds regarding the waste level are set in the system and this information is used along with the storage data in the decision making process in order to generate system commands.

- **Reasoning Layer:** The resources from the ubiquitous layer are fed into the computation unit which inspects the storage data and compares it with the threshold that is set for the dry and wet waste level. The computation unit also records the instantaneous data received at every disposal and provides this information to the monitoring unit to display the waste level timeline along with the individual bin-fill level. On applying logical comparison and evaluating whether or not the threshold is reached, certain topics are created for the AI planner to perform necessary actions through MQTT broker. Express API is used for authentication with respect to GUI login, fetching data, triggering notifications[10] and event-queries.

- **User Layer:** The layer is responsible for delivering application specific services to the user. The monitoring unit summarizes the data to be analyzed by Supervisor on a real time basis[4]. The Supervisor needs to login to the GUI in order to view real time data that is propagated through the system. He/She also gets a notification on Telegram application when required. GUI includes button functionality to check the respective dustbins waste timeline and execute queries.

4 System Implementation

- **System Setup:**
  The waste disposal unit consists of a wet(bin 1) and dry(bin 2) dustbin which are fixed with PIR sensors at the front. When the PIR detects a user, it gets activated which in-turn triggers the servo motor to open the respective dustbin. Both the dustbins are installed with an ultrasonic sensor which computes the level of the garbage; every time a user disposes the waste i.e. the ultrasonic sensor triggers and sends data only when the PIR sensor is triggered. Additionally a load cell is fitted at the bottom of the wet dustbin to measure the wet waste weight and is calibrated according to the weight of the bin placed on top of it. The carrying capacity in terms of weight is considered at the time of pick-up of a full trash bag so that it doesn't rip off and scatter on the surface considering that wet waste is usually heavy in terms of weight. Both of the ultrasonic sensors measure data only when the respective PIR sensors are activated (high) thus resulting in efficient data transfer.

  All the sensors i.e. 2 PIR sensors, 2 ultrasonic sensors, 2 servo motors and 1 load cell all of them are connected to the Arduino UNO micro-controller[1]. Also the ESP8266-NodeMCU is connected to the Arduino UNO as for the sensor data transfer to be sent to the MQTT broker further. The basic

application of the Arduino UNO here is to collect sensor data from various sensors and send it to the ESP8266. The data transfer between Arduino UNO and the ESP8266 takes place by serial communication on two different channels i.e. Arduino UNO to ESP8266 and vice versa because Arduino UNO used here does not support bi-directional serial communication on a single channel. The ESP8266-NodeMCU is connected to a Wi-Fi network to publish the sensor data to the MQTT broker [2]. Also we have connected two LED's to the ESP8266-NodeMCU which are triggered by the AI planner indicating the threshold for the dustbins has been reached for the respective bins. Meeting threshold conditions also impact the servo motor and PIR sensors connected to the Arduino UNO board i.e. the servo motor and the PIR stops working temporarily as not to exceed the waste in the bin.

– **Use of Indirect Communication:**
The indirect communication between the system components happens by publish-subscribe mechanism by using MQTT protocol. The message are published and subscribed with a default QoS of 0. None of the messages are marked as retained as the actuations are supposed to happen when a particular message is subscribed. The data from NodeMCU (ESP8266) is published to Mosquitto MQTT broker using PubSubClient [2]. The GUI server makes use of MQTT packages too in order to exchange and update its input.json file. Server end application publishes a topic to Mosquitto MQTT broker indicating the the bins are full, which is then subscribed by Paho MQTT client in a python script. After necessary actions a topic with different messages will be published to Mosquitto broker. Which is later subscribed by PubSubClient in NodeMCU (ESP8266) and trigger necessary actuations. The Mosquitto project provides a C library for implementing MQTT clients and operate via commands: mosquitto_pub and mosquitto_sub i.e. command line MQTT clients.

– **Threshold Evaluation:**
The threshold of the dry and wet dustbins are graduated by first estimating the completed length of the dustbin without any waste and for the wet waste dustbin; it is the length of the dustbin along with the weight of garbage. These values are considered as maximum values that are recorded on the server. The values were in "cm" for ultrasonic sensors and "kg" for load cell. We compute the percentage of waste level for display on the GUI for better visualization. The threshold is set based on the bin capacity for dry waste and on the decomposition rate for the wet waste. Threshold values pertaining to bin 1 is 2kg for weight and 50% for ultrasonic-fill level whereas for bin 2, it is 70% ultrasonic-fill level. The ultrasonic-fill level of wet waste is kept lower than that of the dry waste is because wet waste tends to decompose quicker resulting in unpleasant stench in the areas surrounding the disposal unit. As

a result, the threshold of bin 1 will be reached earlier for quicker collection of trash. For each of the threshold calculations, we have set certain tolerance value of around 2-3% since the servo motors when operational, vibrate the ultrasonic sensors providing negative percentage levels on the GUI.

– **Automated Planning and Scheduling (AI planning):**
In our project we have used Planning Domain Definition Language (PDDL) as an artificial intelligence planning language[12]. It makes use of certain course of actions that help to transform a given state into a state which satisfies a predefined set of goals and realizes strategies or action sequences, typically for execution by intelligent agents. Two problems files are written with different initial states. When wet bin threshold is reached, a problem.pddl file with wet bin level being higher than the threshold is passed to AI planner whereas when the dry bin threshold is reached another problem.pddl file with dry bin level higher than the threshold is passed to AI Planner which is classical planner as a web service. Different actions are written in domain.pddl file to take different threshold values. If the threshold is reached in either of the dustbins i.e. if a high is delivered in any of the measurements; a topic will be published to Mosquitto MQTT broker. A python script is written which subscribes to the broker for the published topic with Paho MQTT client. The script is written in such a way that, when a message is received that either bin1 (wet) or bin2 (dry) is full, the corresponding prodlem.pddl file will be posted to the AI planning solver. Two different plan.txt files will be generated to perform two different actions. The plan.txt file is read for particular key words and separate message will be published to MQTT broker. The NodeMCU (ESP8266) subscribes to the topic with PubSub client in order to turn ON the corresponding LED and shut down the servo motor once the message is received.

– **Web Application and Notification service:**
The web application has been developed using HTML, CSS and Javascript[5] whereas the web APIs are developed in a framework called Express JS (Express is a web application framework for Node.js). Node.js being an open-source, cross-platform, JavaScript runtime environment helps to execute JavaScript code outside of a browser. It is fundamentally used for event-handling servers and back-end API services.

The server listens to a particular topic (sensor data) and will get the message with the help of MQTT package[7]. This package is event-driven and so it will continue to listen and wait until the message is received. The received message is further updated into the input-json file. In case, there is a new value of an entity, it will be rewritten in association with the updated values (waste levels that keep increasing).

The GUI uses Fetch API[3]; which is a method in JavaScript used to make http requests to/from the server every subsequent second. As a result, data gets collected from input.json and gets sent back to the GUI as a response thus displaying graphs and texts.

The server also checks for received threshold via MQTT and will trigger a telegram message only when threshold conditions are met. The notifications include an alert when the respective bins are full (threshold is reached). Such arrangements help to create awareness regarding the disposal unit without actually having to visit the disposal zone and still schedule timely pick-up of trash. Another trigger mechanism is with respect to the bin opening functionality present on the GUI. Here the GUI makes a request to the server using Fetch API where messages are conveyed to the Arduino via MQTT package. This activation that is done by pressing a button, signals the Broker to switch ON the motor and open the lid of the desired dustbin with its LED turned OFF.

Express helps in fast-tracking development of server-based applications such that there is a bridge between the front-end and back-end. Routes used help to divert users to different section of the web applications based on appropriate requests. There will be one route to reveal the main page and other one to reveal the login page. In login page, credentials will be sent to server via Fetch API where it will be validated with credentials that are already present in input.json. In case the validation fails, it displays "user not found" or "invalid password"as a response. If it is validated then user-key will be fetched from input.json which will be used in a protected route which will verify the input token and redirect you to main page[6]. Once within the GUI, the Supervisor can view the waste levels, weight and waste timeline for each of the dustbins as indicated by the server.

The link to our GitHub repository:
https://github.com/Priyadarshini05/Group9/tree/master

---

5 Discussion and Conclusion with Future Scope

We have entered an era in which IoT is gaining recognition due to its remarkable automation, large data analytics and agglomeration of various technologies working together to provide desired outcome. We therefore need to be aware of its concepts and understand the impact it will have on our lifestyle and various other sectors such as agriculture, medical etc. where time and money are constraints. As mentioned earlier, IoT has its data transferred over a network thus reducing human-human or human-computer interactions since devices talk directly to each other, can make joint decisions and translate data between devices with or without a need for cloud or servers.

With our proposed system, we plan to minimize human efforts and time in regards to waste monitoring and disposal so that multiple and unwanted trips can be avoided by the Supervisor. The waste should only be collected when the Supervisor is notified by the smart waste system and not otherwise. It prevents over-flowing of garbage from the waste bins and maintains an odorless environment since timely collection will be ensured by the waste management system. Users too will understand the importance of waste separation at the initial stage and will be aware of the availability of dustbins to discard waste.

This project can be further extended to multiple floors in a building to which the Supervisor will have access to; through the web application. A mobile application could be developed to inform users on as to where the dustbins are located near to them and which of those dustbins are full or empty. The nearest empty bin could be highlighted for disposal and users will be made aware of the concept of reduce, reuse and recycle. Our project is aimed at implementing smart waste management system in eateries/cafeteria but this can also be installed in schools, colleges, supermarkets and many other places. With reference to the results generated by the proposed system, we can analyze the data to improve and maintain the waste management system in a preferable way such as providing higher capacity bins, minimizing the cost and adding on to the existing proposed system with a number of other dustbins to dispose off other forms of waste materials apart from dry and wet waste classification. Depending on the amount of waste generated on a regular basis, we get the idea of food consumption in eateries that have such smart waste management systems installed.

References

1. Arduino forum, https://forum.arduino.cc/
2. Esp8266: Connecting to mqtt broker, https://techtutorialsx.com/2017/04/09/esp8266-connecting-to-mqtt-broker/
3. Fetch api, https://developer.mozilla.org/en-US/docs/Web/API/Fetch$_API$
4. Introduction.chart.js, https://www.chartjs.org/docs/latest/
5. Javascript, https://developer.mozilla.org/en-US/docs/Web/JavaScript
6. Json web token, https://www.npmjs.com/package/jsonwebtoken
7. Mqtt, https://www.npmjs.com/package/mqtt
8. Nodemcu setup with arduino-ide, https://www.instructables.com/id/Setup-NodeMCU-Programming-With-Arduino-IDE/
9. Smart waste bin, https://create.arduino.cc/projecthub/kmutt-thailand-students-in-training-period-at-esiee-amiens-france/smart-waste-bin-e70fb1
10. Telegram bot api, https://core.telegram.org/bots/api
11. Visual paradigm, https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/
12. Writing planning domains and problems in pddl, http://users.cecs.anu.edu.au/ patrik/pddlman/writing.html
13. Aleena V.J.*, Kavya Balakrishnan, R.T.S.K.K.S.S.T.S.: Automatic waste segregator and monitoring system. www.stmjournals.com pp. 2–4 (2016)